

Idaho National Engineering and Environmental Laboratory

Java XMGR

***Presented by
Steven P. Miller***

***2004 RELAP5 International
Users Seminar***

August 25-27, 2004

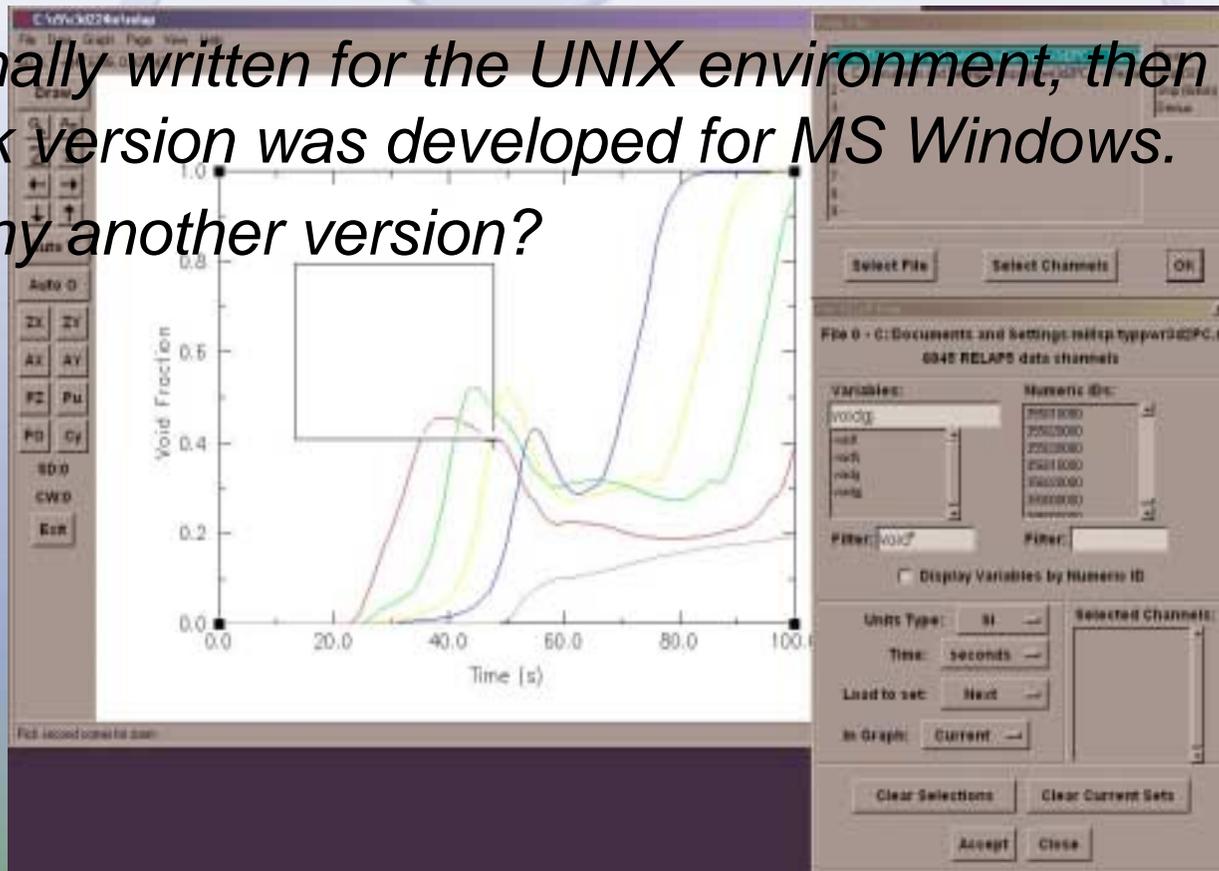


Outline

- ***XMGR5***
- ***Reasons for Conversion***
- ***Conversion to Java***
- ***Details of Project***
- ***Progress***
- ***“Improvements”***
- ***Future Work***

XMGR5

- *XMGR5 is a popular XY plotting utility packaged with RELAP5 then with RELAP5-3D®*
- *Originally written for the UNIX environment, then a Tcl/Tk version was developed for MS Windows.*
- *So why another version?*



XMGR5 - Drawbacks

- *The original XMGR5 does not run natively on MS-Windows Platforms.*
- *This version had to be compiled on every computer running it.*
- *This complicates the distribution of the XMGR5 product with license issues.*
- *As the operating systems evolve and change, these special procedures may have to be upgraded or replaced altogether.*

tkXMGR - Drawbacks

- *The major deficiency of tkXMGR5 is that it is missing some features of XMGR5*
- *Since development of the freeware version of Tcl/Tk has essentially stopped, these problems will not be fixed.*
- *Operating systems continue to change and new ones will be created, the lack of maintenance and development of the Tcl/Tk freeware will eventually cause it to operate incorrectly or fail completely on some future operating systems.*

Reasons for Conversion

- *Portability and licensing were perhaps the most important issues leading to the original choice of Tcl/Tk and now of Java.*
- *Java's popularity has lead to a great deal of development of the language.*
- *Java has a huge number of class libraries for ease of use.*
- *Java is multi-threaded, so it can take advantage of multiprocessor systems.*

Reasons for Conversion

- *The prototype Java RELAP5-3D Graphical User Interface (RGUI), prototype Thermal Hydraulic User Model Builder (THUMB) and SNAP have already been written in Java.*
- *The Java version would be able to run on every platform that supports JRE version 1.4.2 or later.*
- *At this time this includes the following platforms:*
 - *Solaris (SPARC or x86)*
 - *Linux (Intel or IA64)*
 - *Windows (IA64, XP, 2000, ME, NT, 98 or 95)*
 - *Mackintosh (OS 8, 9 or X)*
- *These are simple to install and comes already installed on most new computers.*

High-level considerations

- *Before beginning, input from the principal users of XMGR5 at the INEEL was obtained.*
- *These consultants were determined to be a good representation of worldwide users*
- *This group indicated that they preferred much of the look and feel of the original version of XMGR5 to the current Tcl/Tk version; So I continued with that in mind.*

High-level considerations

- *The following prioritized list of features was developed:*
 - *Read in a RELAP5 file and Draw Graph*
 - *Print Graph (Printer, Postscript, .jpeg, etc.)*
 - *Rescale Plots (on either axis or both)*
 - *Display Sets Status*
 - *Titles (axes, strings arrows, legends)*
 - *Transformations – regression, evaluate expressions (unit conversion)*
 - *Ability to modify Line width, Symbols, Colors, etc.*
 - *Run Scripts*
 - *Arrange multiple graphs on a page (specify columns and rows)*
 - *Ability to set default to auto-scale when reading*

Conversion to Java

- *We chose to retain the C code. This meant that, in many instances, the code would still have to be compiled on each individual workstation.*
- *Therefore, throughout the project, any changes in the original code were made with this in mind.*
- *Standard ANSI functions were used wherever possible to produce the most generic code.*
- *This will reduce future maintenance and development costs.*
- *It should also significantly reduce the time it takes to compile on most computers.*
- *For example, in the current Windows version, only 2 files are required for XMGR to run and no on-site compiling is necessary.*

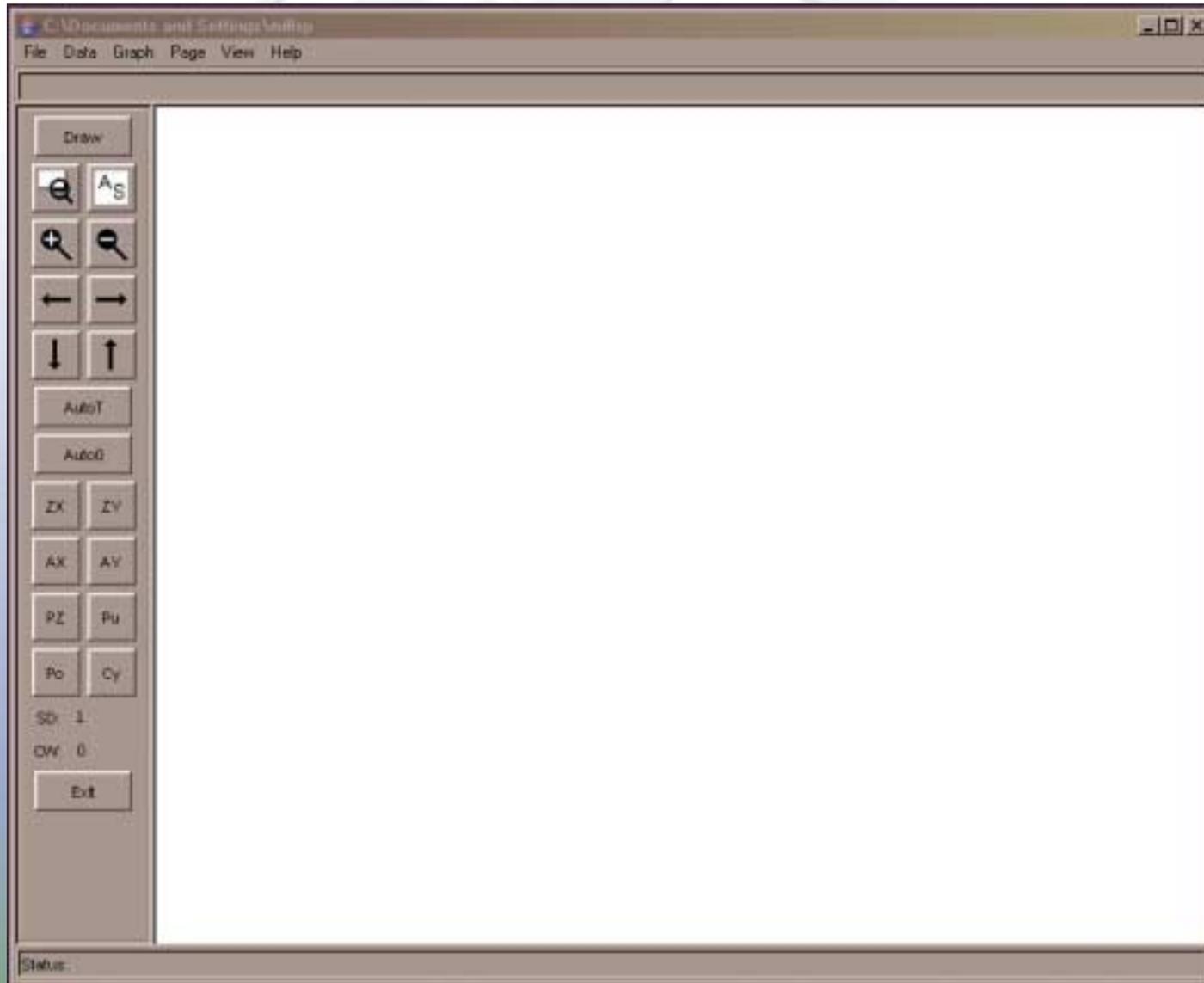
Conversion to Java

- *The ultimate goal of the Java conversion project would be a completely Java version of XMGR.*
- *A complete rewrite would have to start at the algorithmic level.*
- *That is beyond the scope of this task*

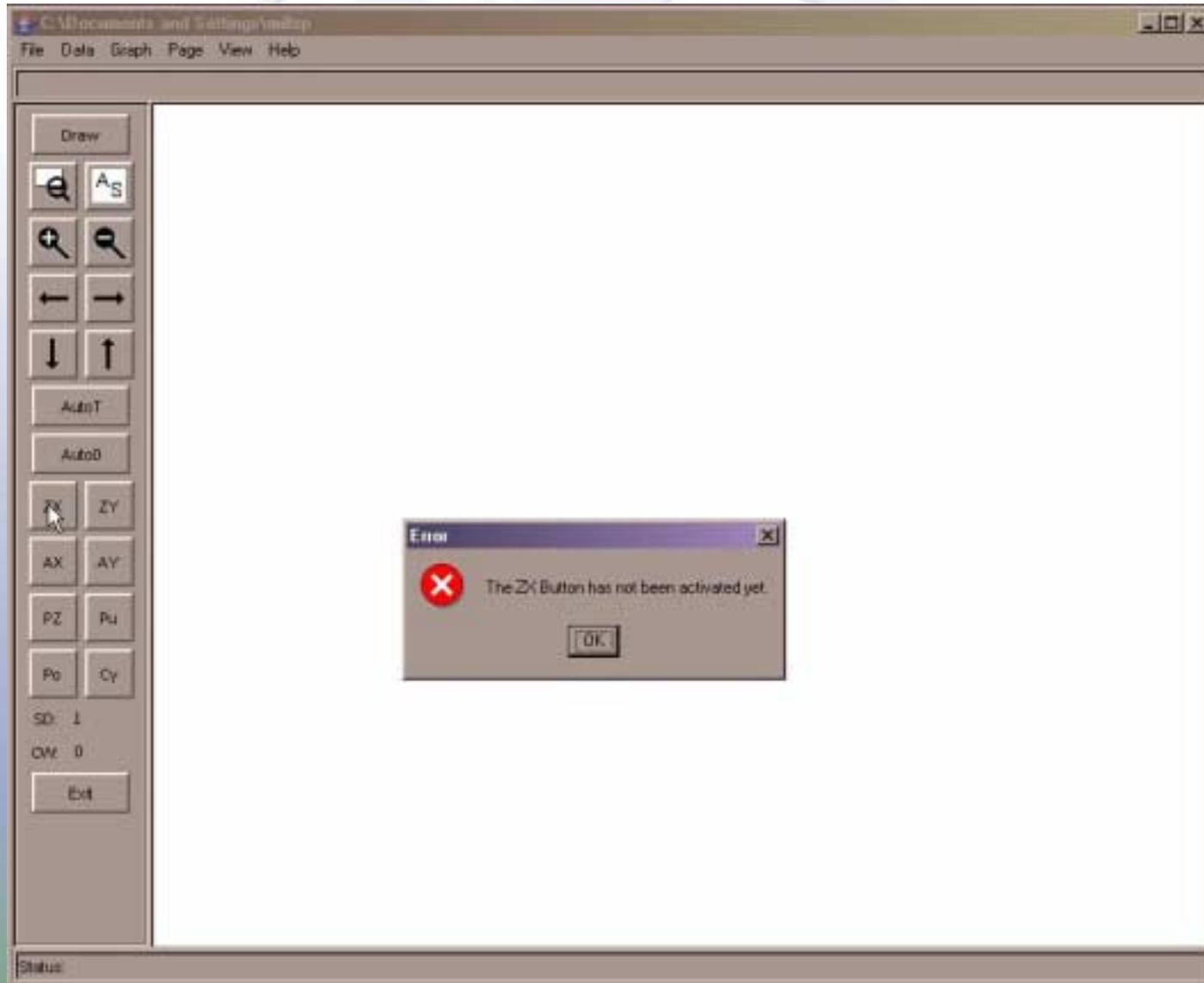
Details of Work

- *The work of creating Java XMGR5 then can be broken into two parts, the infrastructure work and the work of activating XMGR5 features. The activation work is reported in Section 4.0. The infrastructure work was broken into four distinct tasks or steps. These are:*
 - *Build the base GUI*
 - *Establish communication from Java to the C procedures of XMGR5*
 - *Establish communication from the C procedures of XMGR5 back to Java*
 - *Extract data from a RELAP5-3D restart-plot file*

- *First the base GUI had to be built.*



- *It was visually pleasing, but nothing worked.*





Details of Work

- *Because of JNI, functions took considerably less time then to develop than functions from Tcl/Tk to C;*
- *There is no JNI equivalent so one has to create Tcl/Tk commands and “register” them to accomplish the functionality that JNI supplies.*
- *Finally a successful communication method was found, namely, passing all the necessary parameters to each function involved in every process that involved the C side talking back to the Java side.*
- *This meant changing 300+ functions to include the necessary Java parameters.*
- *This was very time consuming, but once this was finished, communication from C to Java worked.*

Details of Work

- *The next task was to read in a RELAP5 restart-plot file.*
- *This involved two steps:*
 - *The first was to read the restart-plot file and place the data into memory*
 - *The second was to access a list of XMGR5 channels; this is actually a list of data headers.*

Progress

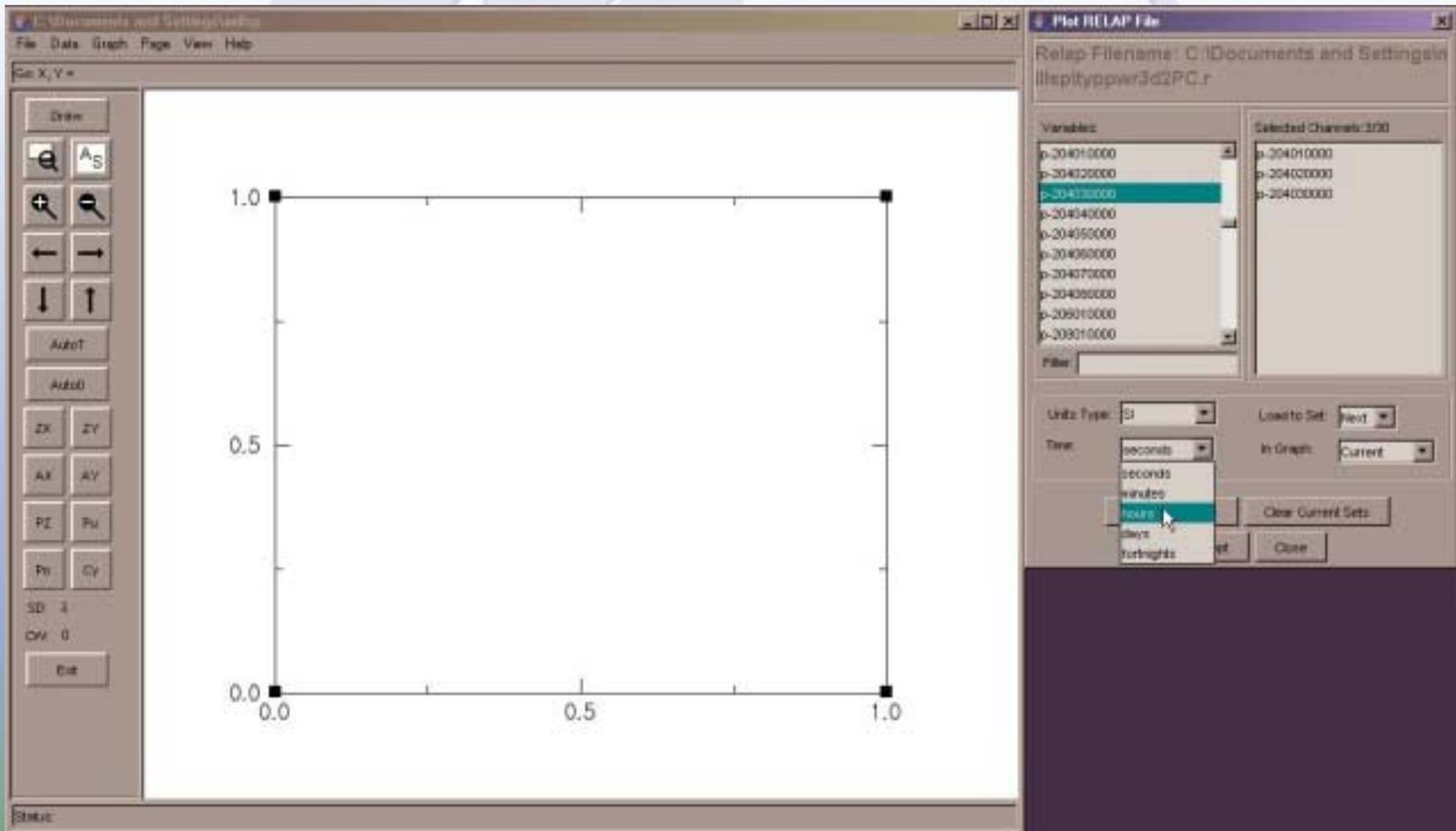
- *It then became possible to:*
 - *Hook up functionality to the new interface*
 - *Retrieve data from a RELAP5 restart-plot file*
 - *Perform calculations with it*
- *It was then be possible to display the data.*
- *Then it was necessary to write both the C code to communicate with the appropriate Java code*
- *Finally write the Java code that would display the data on the screen.*

Progress

- *So far Java XMGR the following has been completed:*
 - *Automatically save settings (working directory, last file used, screen size and placement) on exit.*
 - *Automatically resize the plotting panel.*
 - *Show existing fonts (e.g. Cyrillic).*
 - *Move the graph up, down, left, and right.*

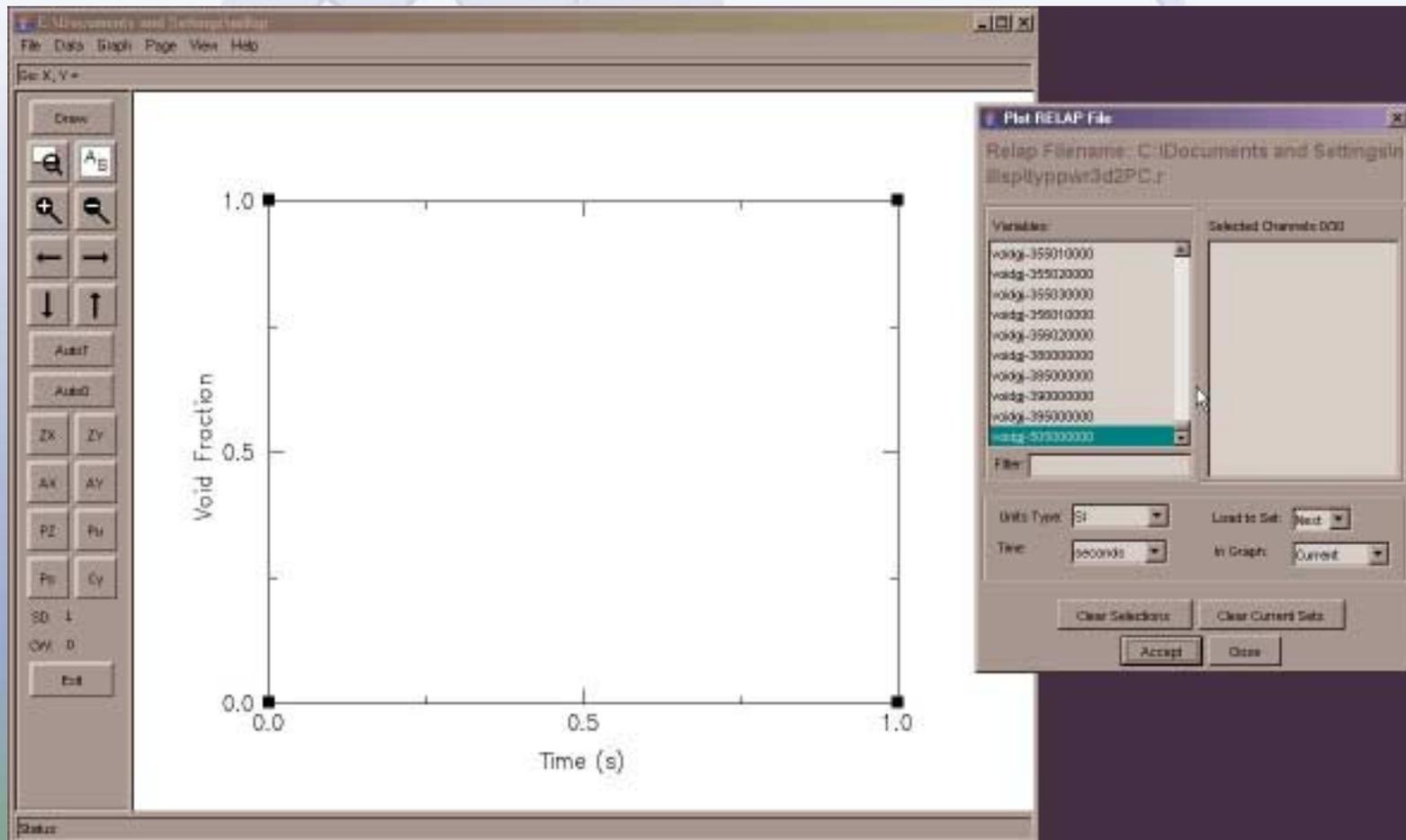
Progress

- Read in a RELAP5 restart file and choose which channels to plot.



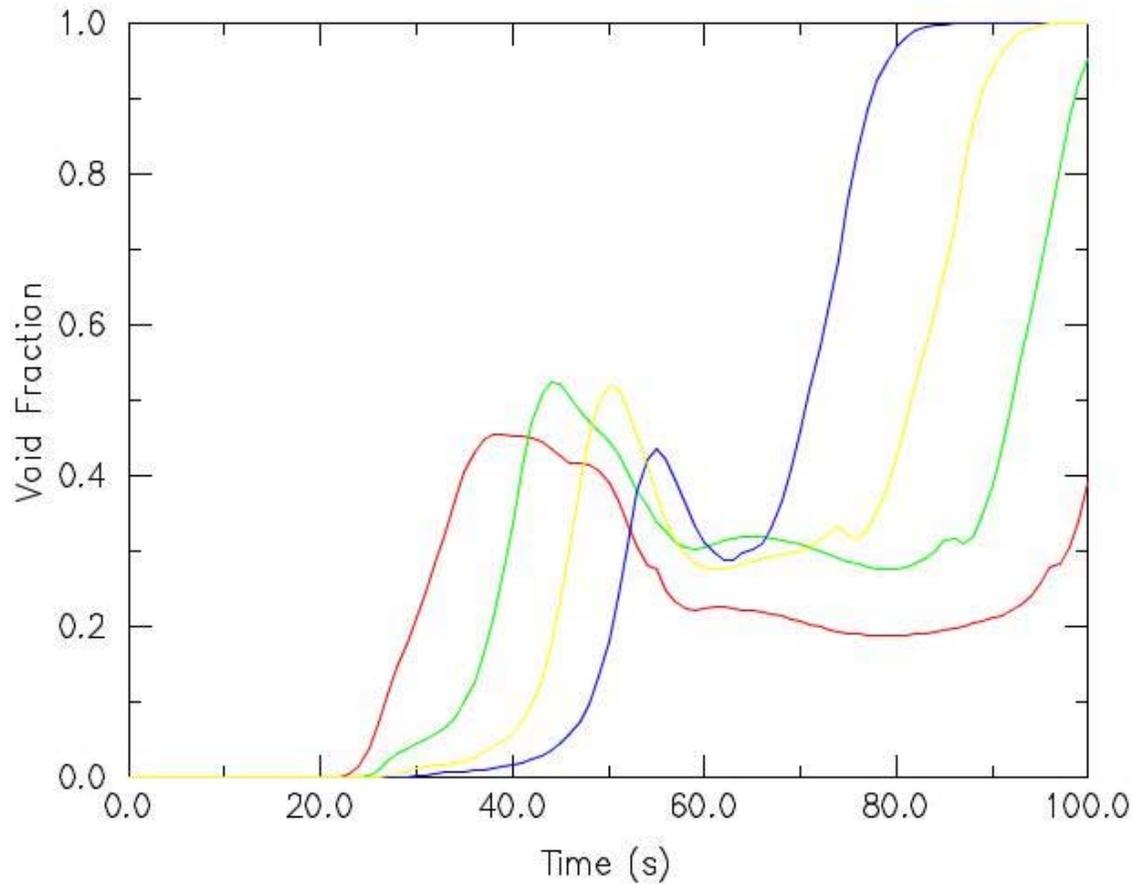
Progress

- Plot a graph from a RELAP5 restart file.



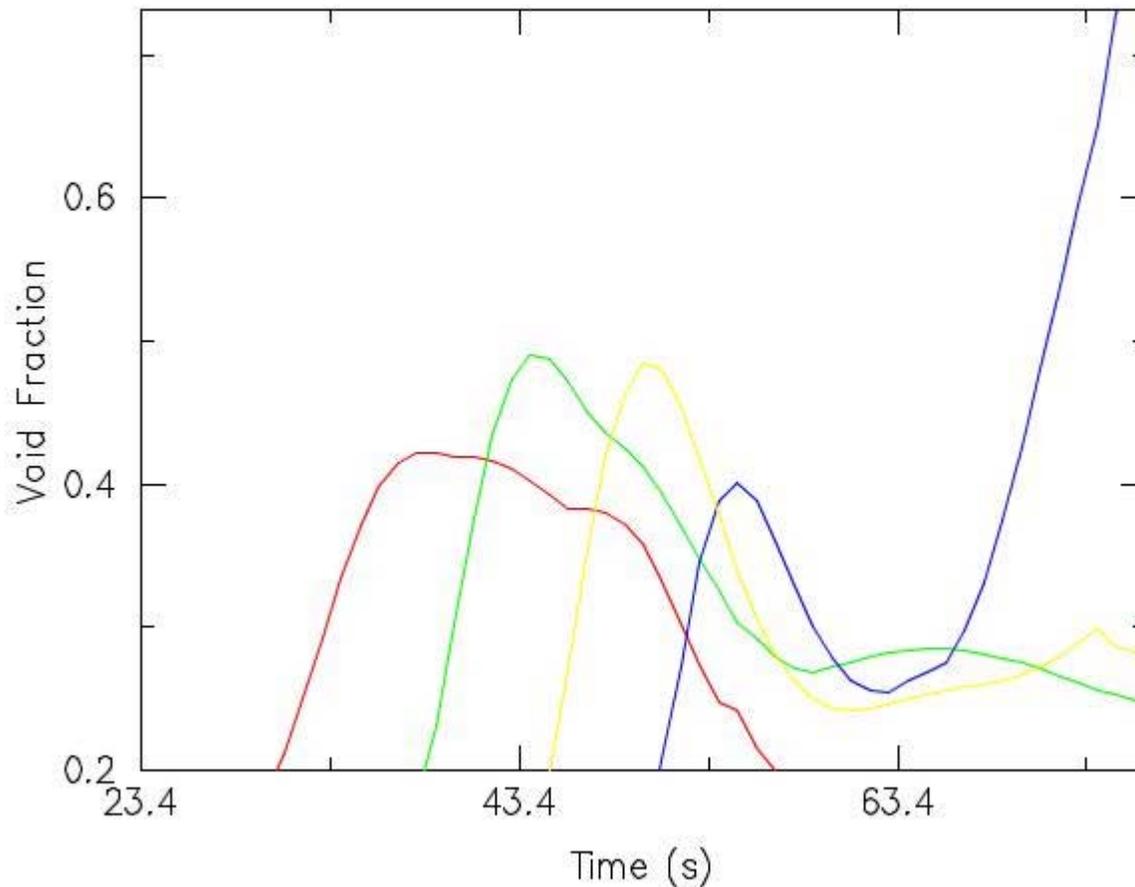
Progress

– Auto scale the graph.



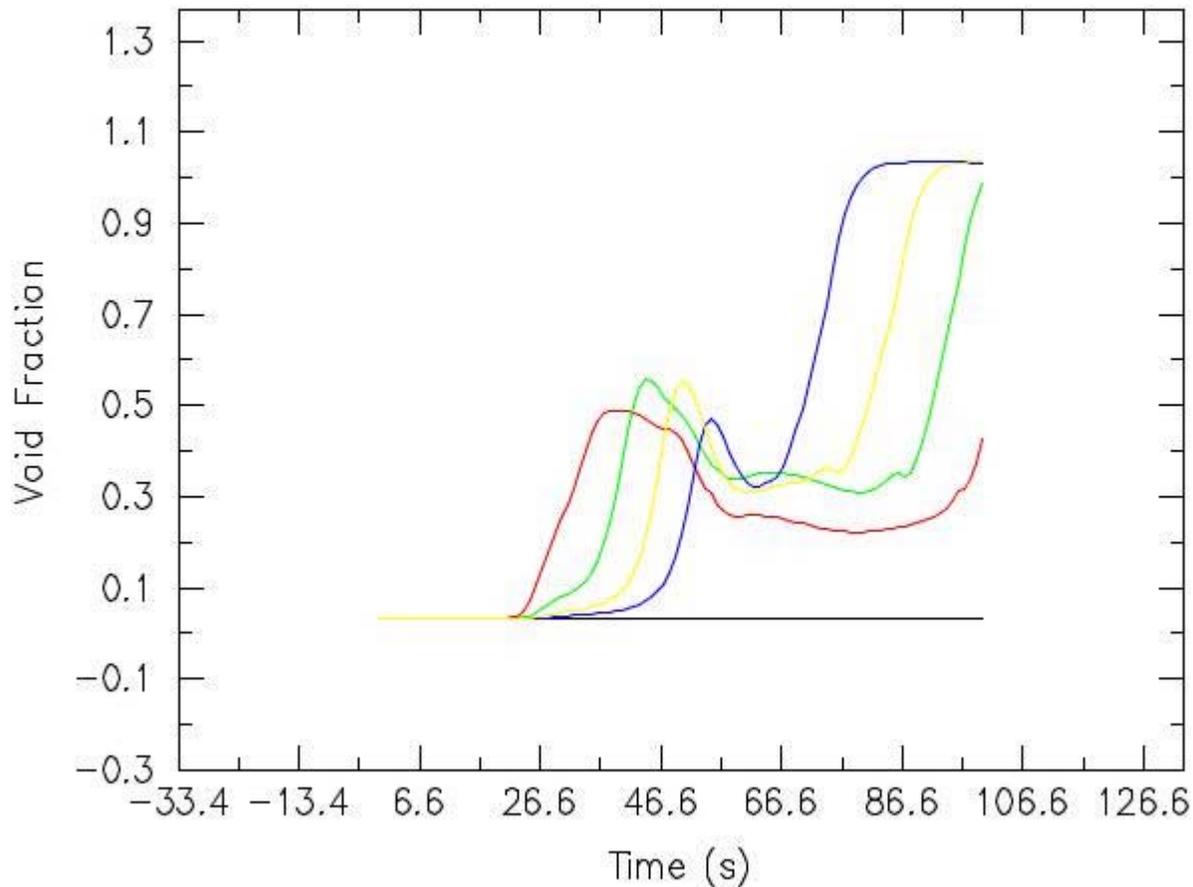
Progress

– Zoom in on the graph.



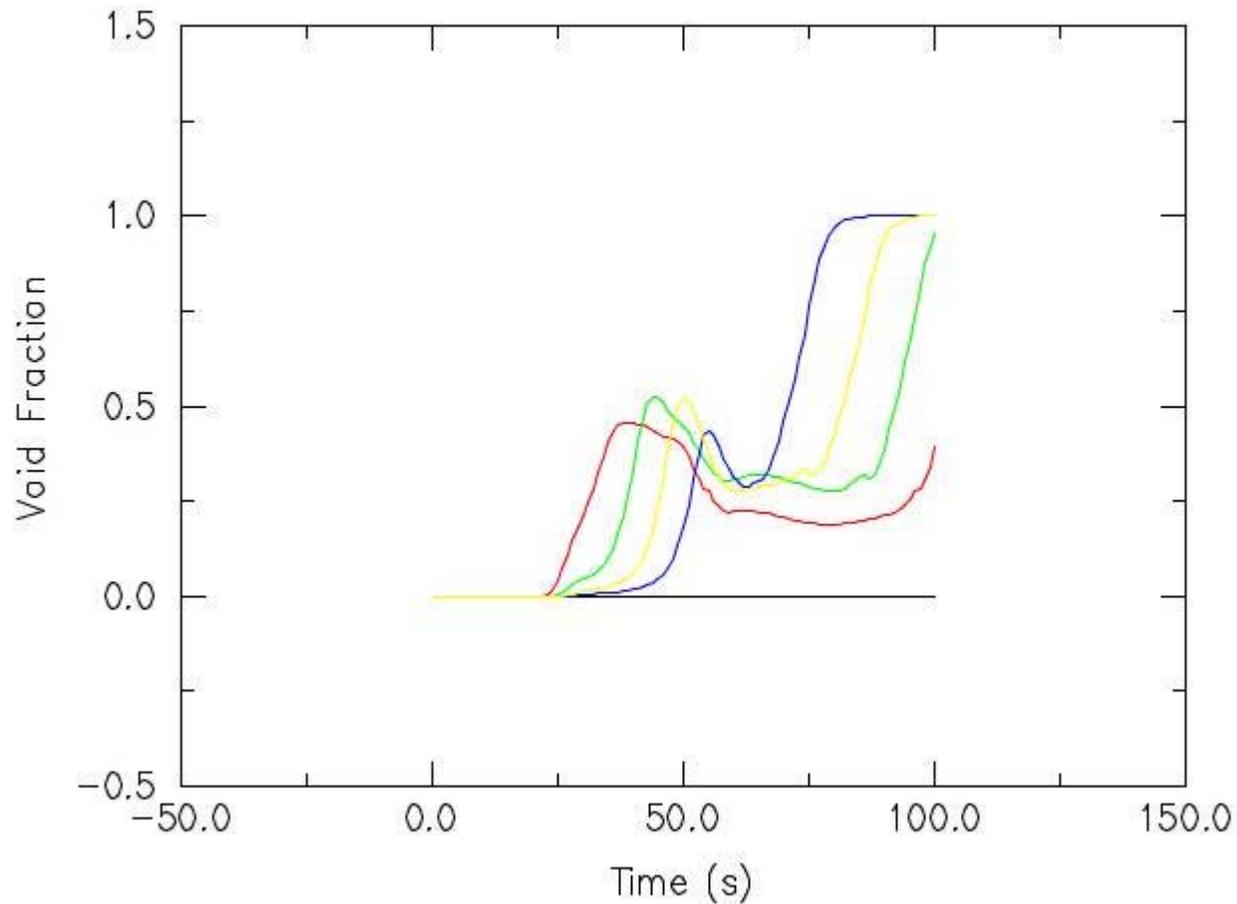
Progress

– Zoom out on the graph.



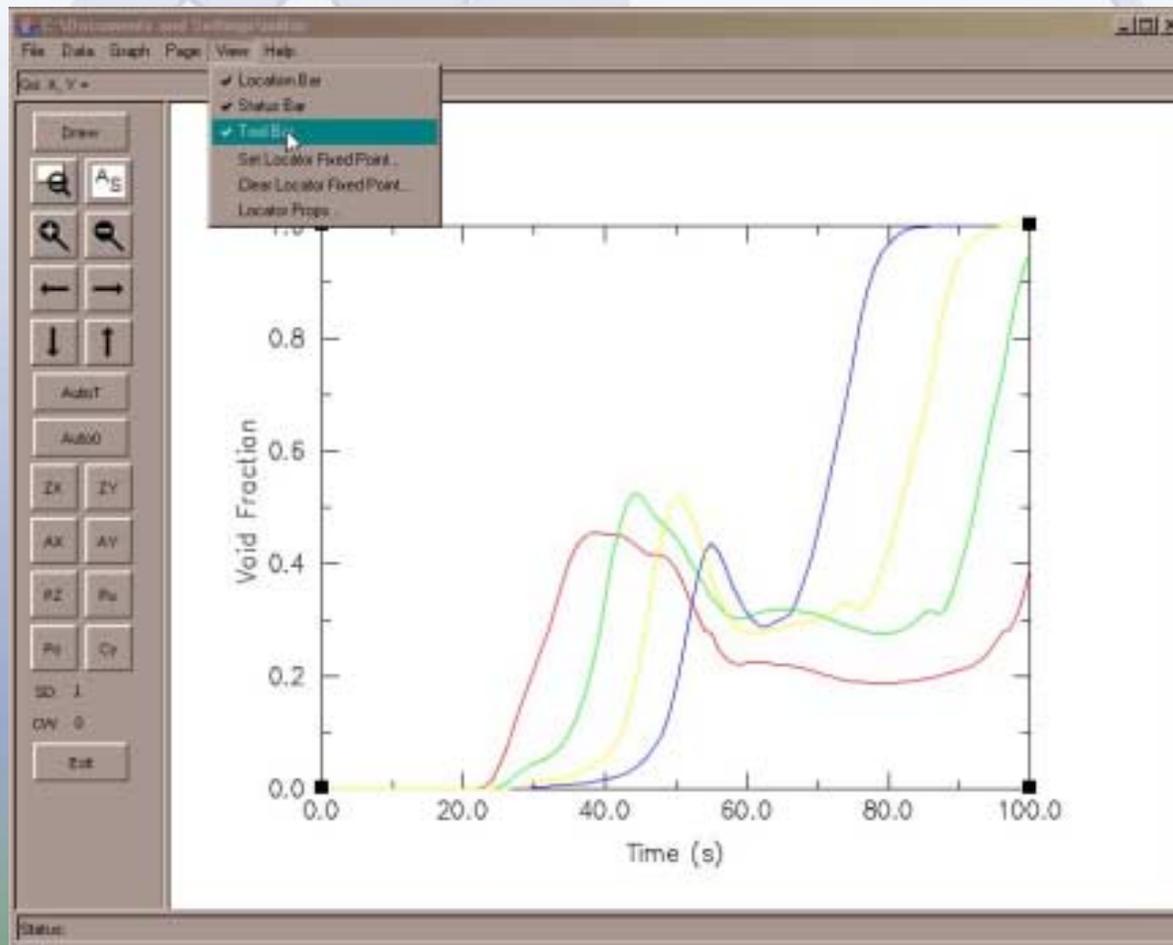
Progress

– *Set Auto-Ticks.*



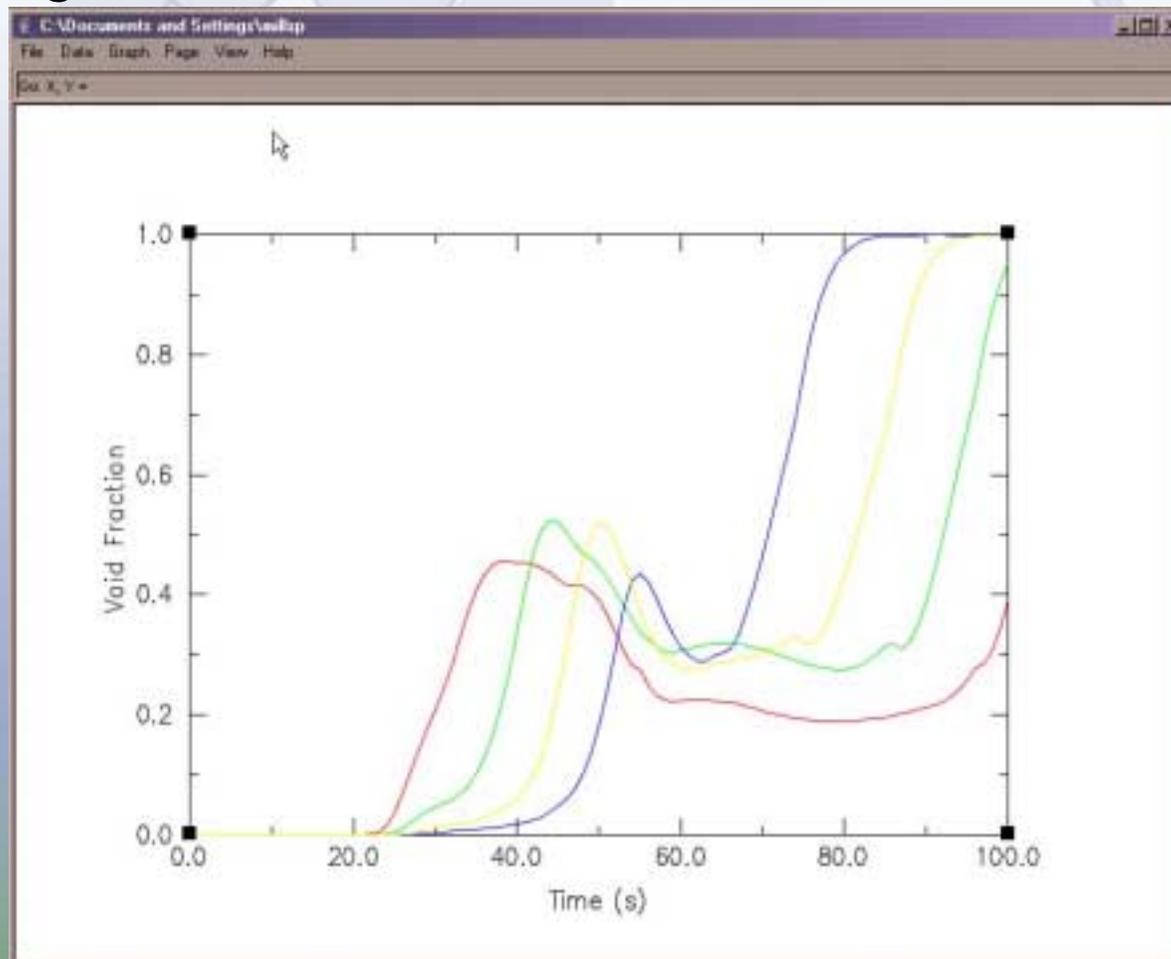
Progress

– *Change the view.*



Progress

– *Change the view.*



Progress

- Read status of sets, graphs, variables, and regions.

set#	n	stat	type X/Y	min	at	max	at	mean	std. dev	comment
0	101	on	XY X Y	0	1	100	101	50	29.155	voidgj-355...
1		De-activated				4.173e-005	80	5.5323e-0...	1.1883e-005	
2	101	on	XY X Y	0	1	100	101	50	29.155	voidgj-355...
3	101	on	XY X Y	0	1	100	101	50	29.155	voidgj-356...
4	101	on	XY X Y	0	1	100	101	50	29.155	voidgj-356...
5		De-activated				0.95204	101	0.25859	0.22684	
6		Undefined				0.9998	92	0.35095	0.40101	
7		Undefined				0.99976	101	0.29965	0.32448	

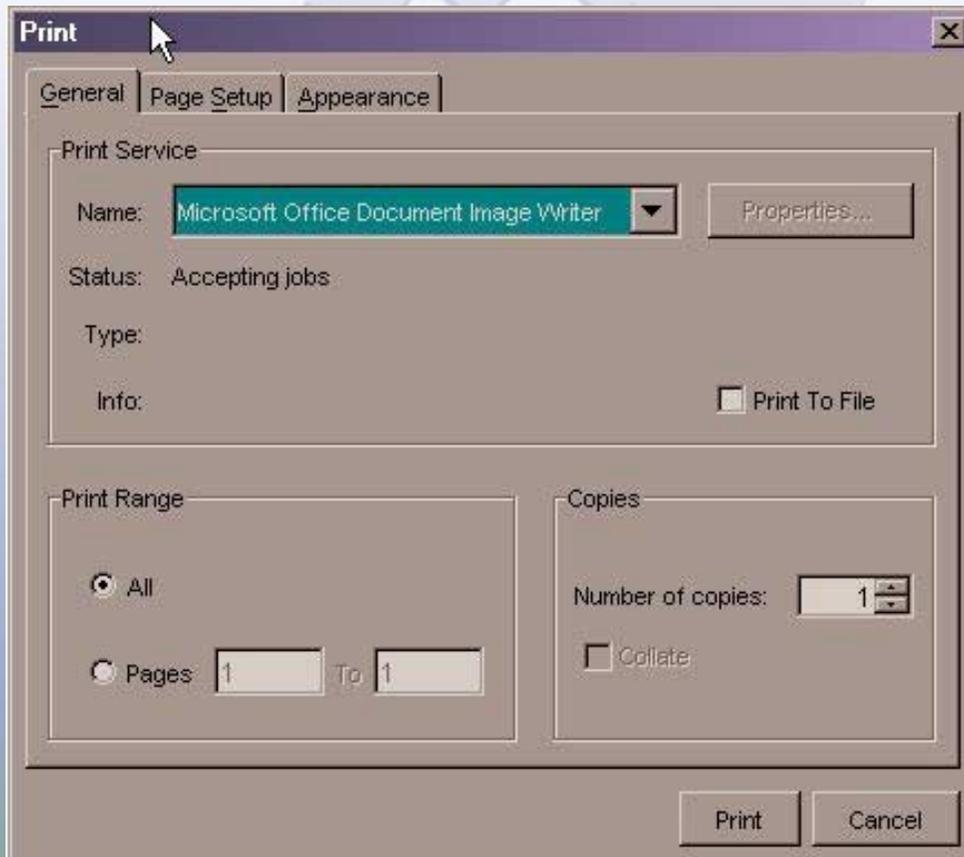
Click on the set index number to kill the set

Buttons: Kill, Deact, React, Copy, Move, Auto, Reverse, Join, Pack, Cancel

Buttons: Close, Update, Write, Page+, Page-, Home, End, Display: Sets

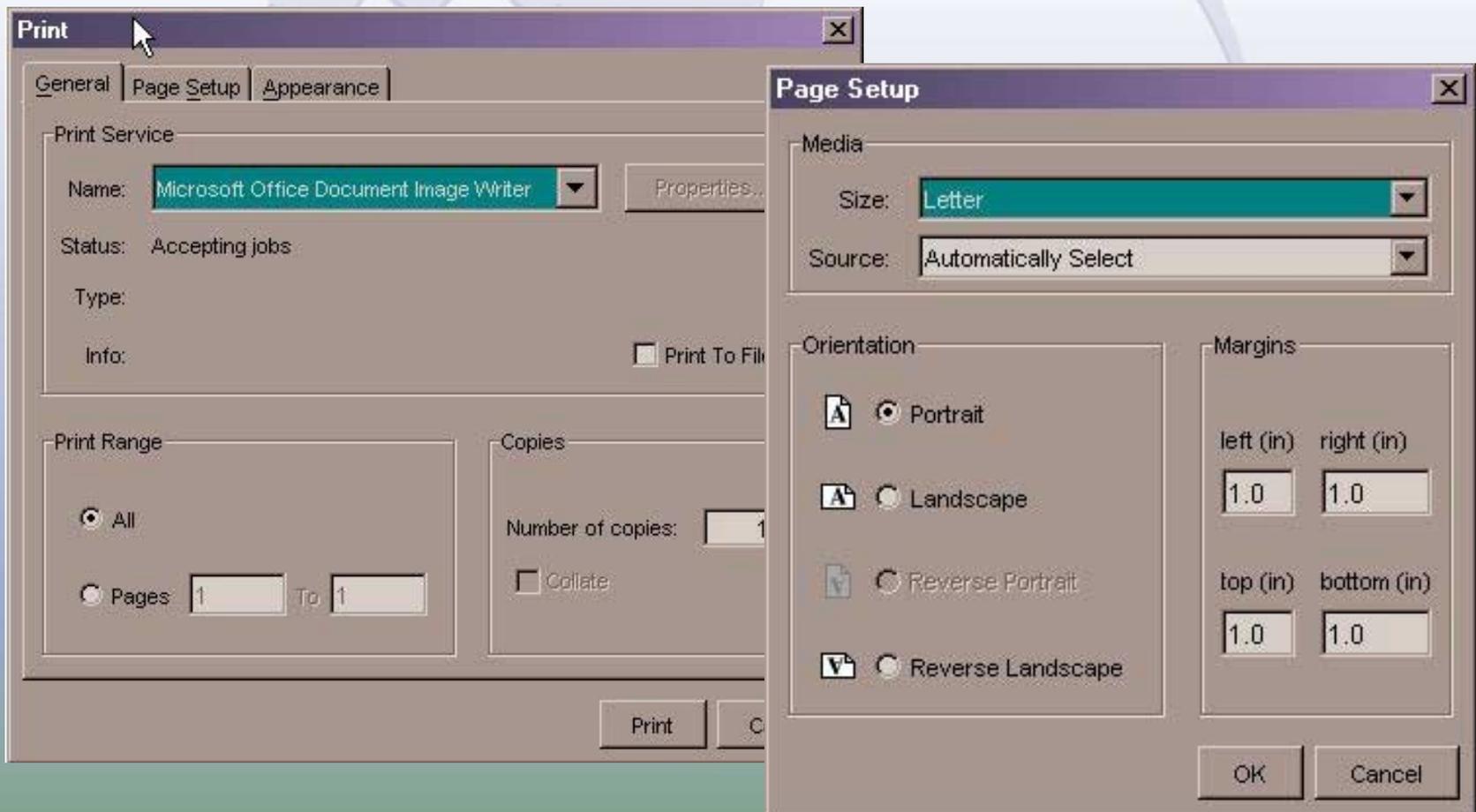
“Improvements”

– *Uses native environment for Print functions*



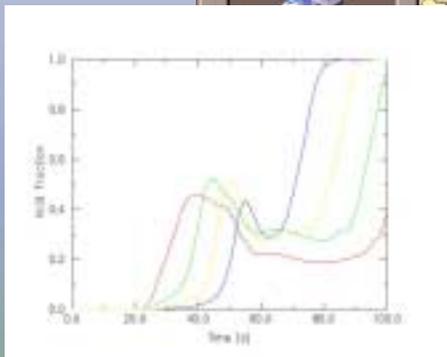
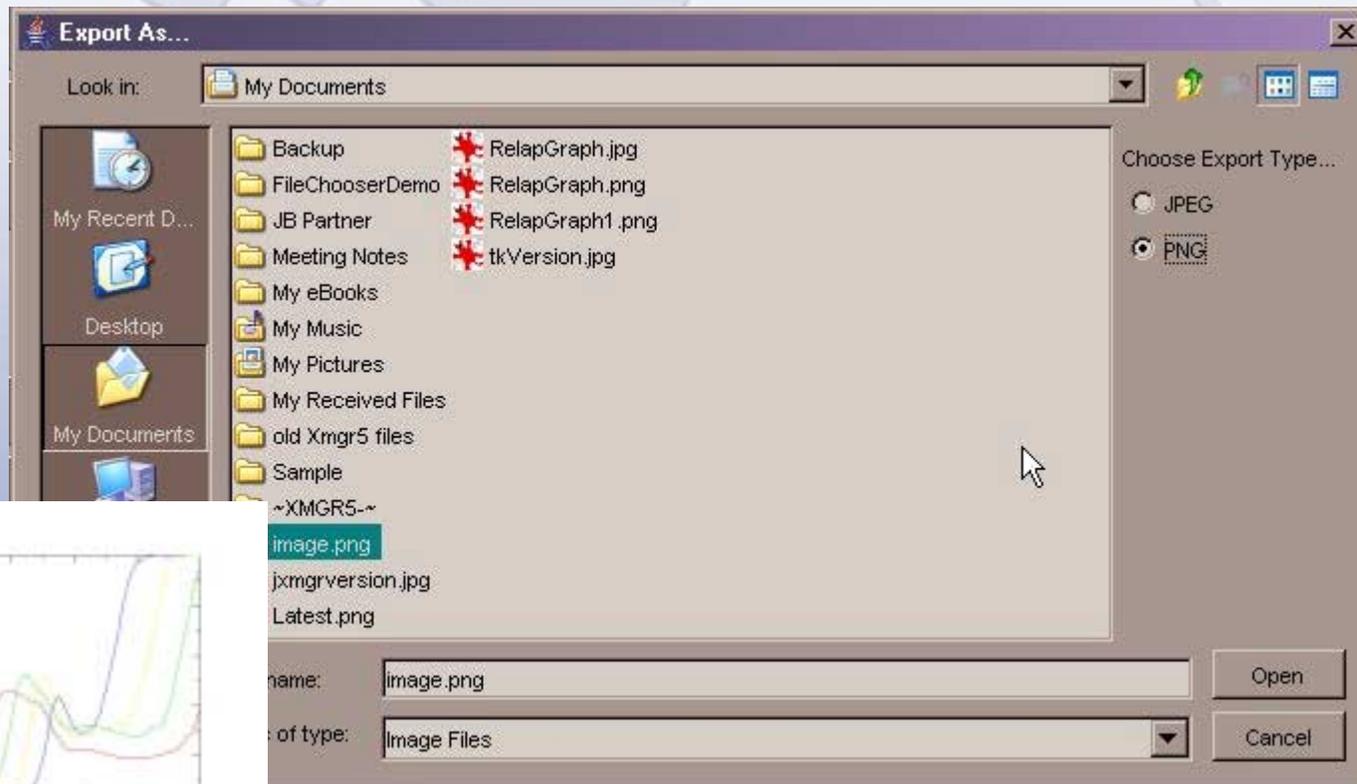
“Improvements”

– And for Page Setup



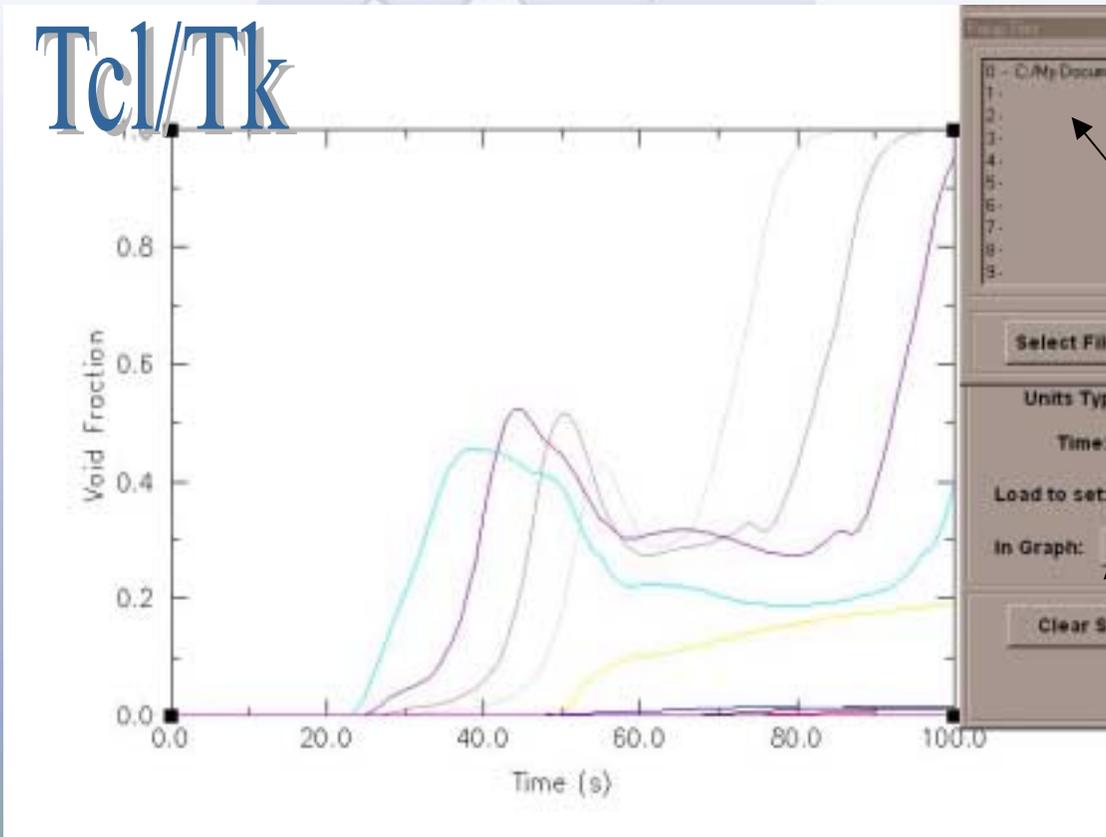
“Improvements”

– *Export an image.*



“Improvements”

- Improved graphic output

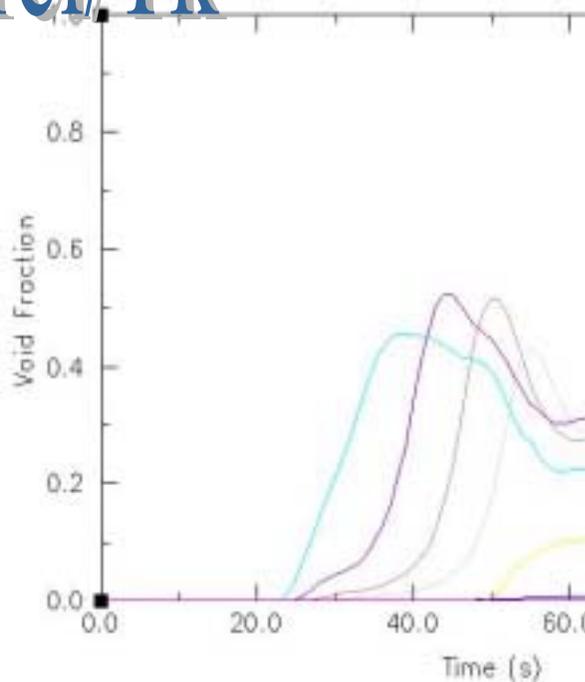


*Captures
pieces of
unwanted
objects*

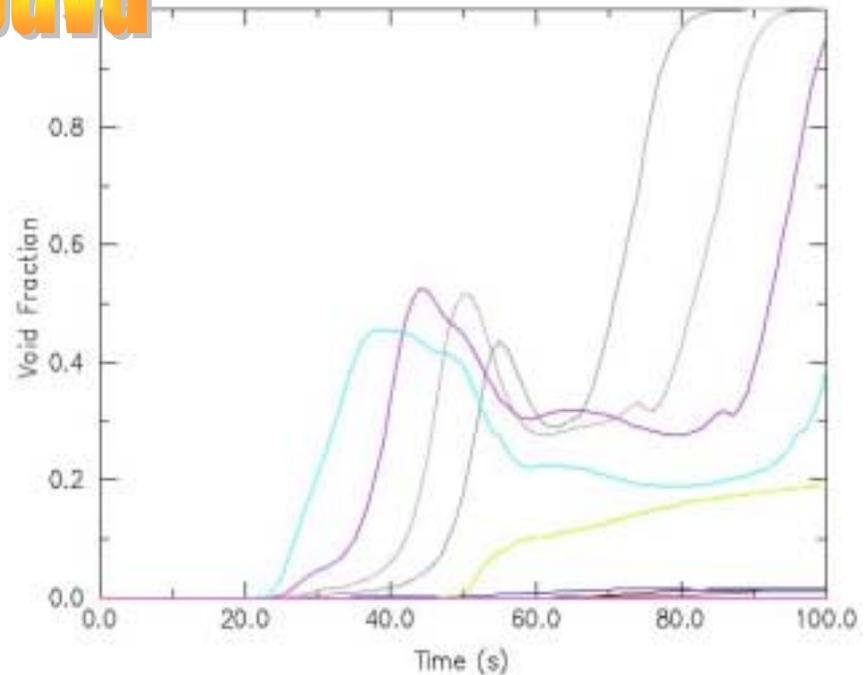
“Improvements”

- Improved graphic output

Tcl/Tk

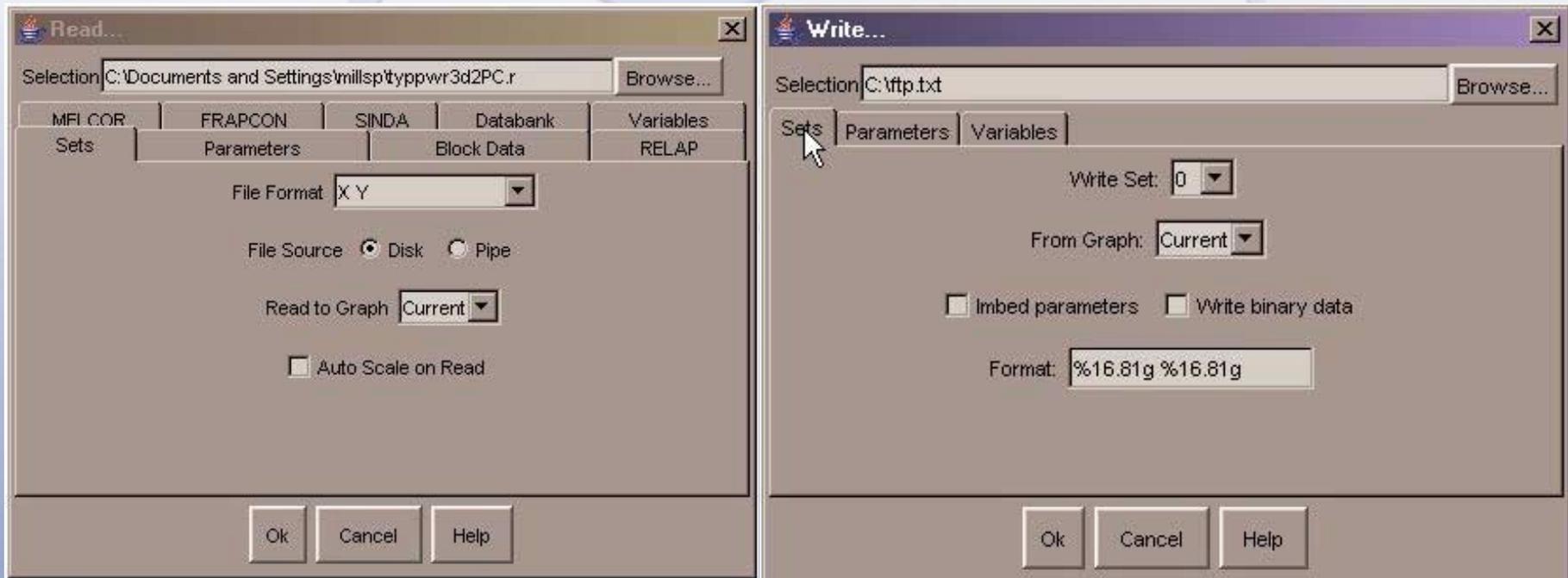


Java



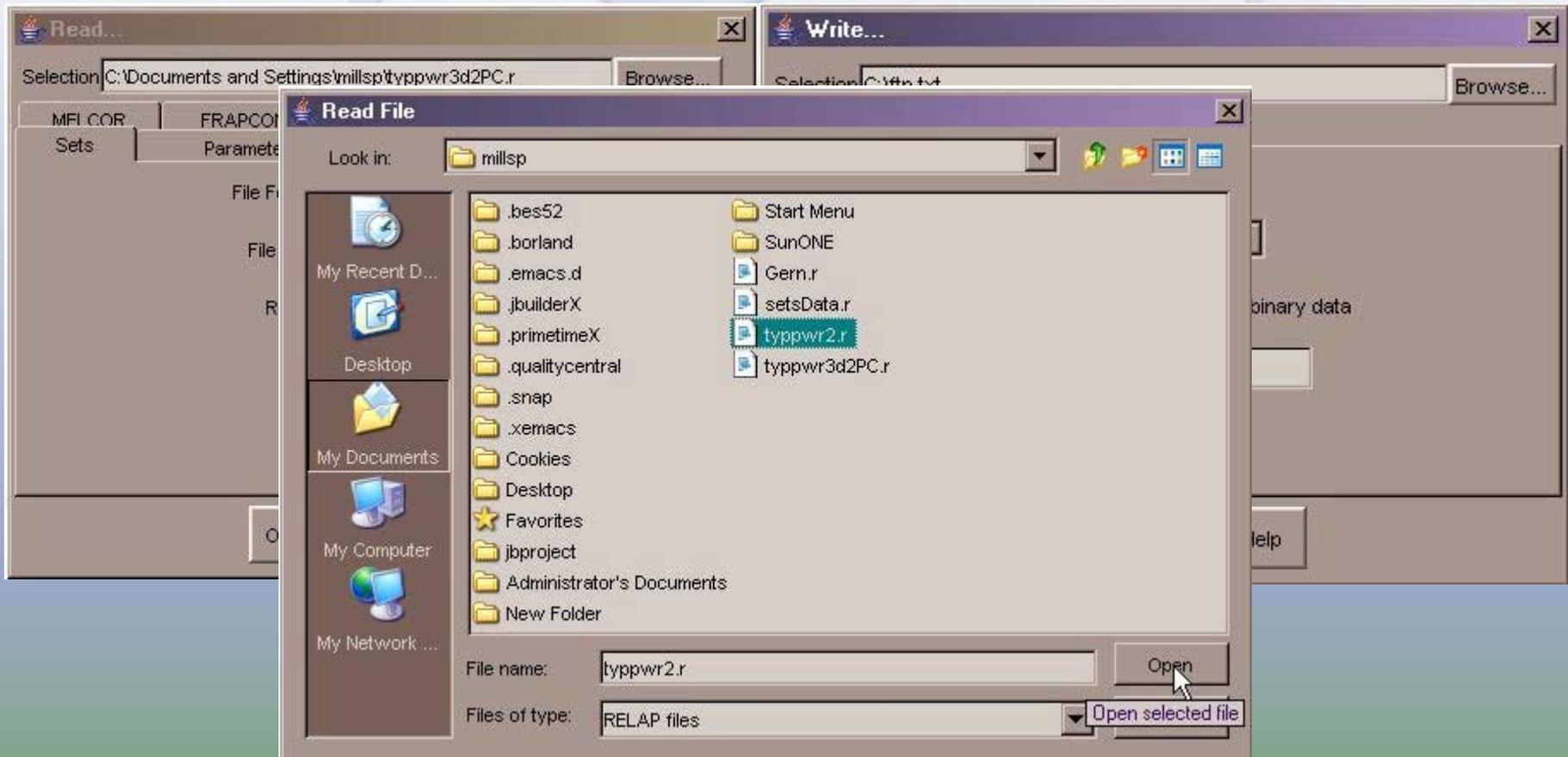
“Improvements”

- *The Read/Write dialog is now tabbed for convenience*



“Improvements”

- *The File-Chooser uses native file readers*



The Future

- *What's left?*
 - *The work that must be performed to complete the project includes completion of all items is listed earlier.*
 - *Many of these items have a number of subtasks associated with them.*
 - *testing it on numerous operating systems*
 - *preparation for future transmission with RELAP5-3D product releases*
 - *Writing appropriate documentation.*
 - *There are additional features of XMGR5 that are not listed that would also be desirable to activate and some or all of these may become future work scope.*